REMARKS

In the Final Office Action dated April 16, 2004, claims 1-20 are pending. Claims 1, 8, and 11 are independent claims from which all other claims depend therefrom. Claims 1, 8, and 11 have been amended. Note that claims 1 and 11 have been amended for clarification reasons not for patentability reasons. Also note that claim 8 has been amended to further distinguish it from the Russell reference cited below, although Applicants believe that it is in a condition for allowance without the stated amendment.

Claims 1-7 stand rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, the Final Office Action states the limitation "using a separate controller" recited in lines 9-10 renders the claim indefinite. The Final Office Action further states that it is not clear what function the controller performs and what the controller is claimed as being separate from. Claim 1 has been amended to better clarify what is meant by the use of a separate controller. Note that in the previous Response the Applicants stated that the first four elements of claim 1 are performed separate from the fifth element, as is denoted by the use of a separate controller.

Claim 1 has been amended such that the first four elements, namely: A.) extracting state information from the state model; B.) processing the extracted state information; C.) generating a state code and a state table in response to the processed extracted state information; and D.) compiling the state code to generate a runtime code, are performed using one or more controllers. The fifth element E.) of implementing the state model by running the runtime code while utilizing information within the state table is performed using a designated controller that is separate from the controller(s) used in limitations A-D.

Applicants, respectfully, submit that the limitation E cl arly states the function of the designated controller recited therein. The designated controller recited in limitation E implements or executes the state model. The designated controller in implementing the state model executes the runtime code while utilizing information within the state table. Thus, Applicants submit that the claim 1 is now in a condition for allowance at least with respect to 35 U.S.C. 112. Also, since claims 2-7 depend from claim 1, claims 2-7 are also in a condition for allowance for at least the same reasons.

Claims 1-6, 8, 11-15, and 20 stand rejected under 35 U.S.C. 102(a) as being anticipated by Russell (USPN 6,212,625 B1).

The Final Office Action states that the differences between the limitations of claim 1 and that of Russell are in limitation E of claim 1. The Final Office Action states, which Applicants agree, that Russell does not appear to teach implementing the state model by running the runtime code while utilizing information within the state table using a designated controller. Applicants submit that this limitation is not taught or suggested by Russell.

Russell does not teach or suggest implementing the state model by running the runtime code while utilizing information within the state table nor does Russell teach or suggest performing such implementation using a separate designated controller. The state table of Russell is utilized before the execution of terminating entries and is used to determine entries to execute, whereas the state table of claim 1 is utilized during the execution of a runtime code to perform state transitions. Also, nowhere in Russell is the implementing of a state model using a separate controller taught or suggested, as described above. Russell simply discloses the use of a single execution unit or controller for the execution of a finite state machine. Thus, since each and every limitation of claim 1 is not taught or suggested by Russell, claim 1 is novel, nonobvious, and is in a condition for allowance.

Claim 8 recites a method for implementing a pre-designed plurality of state models for a state machine having an event configuration file. The method includes: i.) extracting state information from the state models; ii.) generating an events symbols header, having global and shared event symbol definitions, from the event configuration file, the event configuration file and said events symbol header defining events to be shared; iii.) processing the extracted state information in response to the events symbols header; iv.) generating state codes and state tables in response to the processed extracted state information; v.) compiling the state codes using the events symbols header to generate runtime codes; and vi.) implementing the state models by running the runtime codes while referring to the state tables.

Applicants submit that Russell does not teach or suggest limitation ii and v above. In the previous Response Applicants submitted that Russell does not teach or suggest the generation of an events symbols header, having global and shared event symbol definitions, from an event configuration file. Applicants submit that the arguments presented in the previous Response at least with respect to this limitation remain and are valid. Applicants further submit that the newly added limitation of the event configuration file and the events symbol header defining events to be shared is also not taught or suggested by Russell.

The events symbol header of claim 8 contains global and shared event symbol definitions. The Final Office Action states that the terms "global" and "shared" are essentially synonyms and that a global or shared variable is one that is used throughout an entire file, program, or system. The Final Office Action infers that the symbol identifiers of Russell are global and shared event symbol definitions, and relies on the ability of Russell to allow for the execution of various state machines independent of hardware implementations for such inference. Applicants submit that in col. 5, lines 16-25, Russell states that the state engine of Russell may be configured to execute any finite state machine.

Since the state engine is capable of executing different finite state machines, execution of the finite state machines is independent of hardware implementation. Applicants submit that this ability is irrelevant with regards to whether the symbol identifiers of Russell are global or shared.

The state engine of Russell may be configured to operate on different hardware devices and may be modified to execute a particular finite state machine on particular hardware device. In general, a state engine may be operated on different hardware devices and symbol identifiers or variables utilized by the engine may be local, global, or shared. Also, whether symbol identifiers or variables utilized are global or shared is immaterial as to whether a state engine can be modified to execute different finite state machines. A state engine can be modified when identifiers and variables are local or global, but the difficulty realized in performing such modification can be quite different. Note that nowhere in Russell are any of the terms "global", "shared", "event symbol header", "events symbol definition" or any synonyms thereof recited.

The symbol identifiers of Russell are not variables nor are they variables that are global or shared. The symbol identifiers, as stated in col. 6, lines 1-7 of Russell, are as denoted. The symbol identifiers are used to identify a current symbol. The symbol identifiers are compared with a current symbol in determining a match entry.

In addition to not teaching or suggesting generation of an events symbols header, having global and shared event symbol definitions, from an event configuration file, Russell also does not teach or suggest the limitation of the event configuration file and the events symbol header defining events to be shared. Not only does Russell not disclose the use of global and shared variables, Russell also does not disclose the use of shared events. In the present application global and shared variables, and shared events are utilized between multiple state models and by multiple controllers. In Russell a single controller

implements a single finite state machine in a sequential format. Thus, Russell is not concerned with the sharing of variables and events, as is the claimed invention.

Furthermore, since Russell does not teach or suggest limitation ii and since limitations iii-vi are performed in response to information contained within the configuration file and the events symbol header, limitation iii-vi are also not taught or suggested by Russell. Also, note that limitation vi in and of itself is not taught or suggested as stated above with respect to claim 1.

In recognizing and utilizing shared or common events between state models the claimed invention of claim 8 provides an easy technique for modifying existing state machines and versatility in the design and implementation of multiple state models. This is clearly not taught or suggested by Russell. Thus, Russell also does not teach or suggest each and every element of claim 8, therefore, claim 8 is also novel, nonobvious, and is in a condition for allowance.

Claim 11 recites a state processor for generating a state table and a runtime code for use in the implementation of one or more pre-designed state models <u>having hierarchial organized states</u>. The processor of claim 11 includes a state model information provider that extracts state model information <u>in response to the state models</u>. A state information separator generates a state code and the state table <u>in response to the state models</u>. A compiler compiles the state code and generates the runtime code. The terms "hierarchial organized states" refers to the ranking and order of states within the state models. Russell traverses through a sequence of states for a finite state machine. Russell does not provide a ranking between states, but rather simply transitions through a predetermined order of states. The ranking between states not only provides a ranking between states of a single state model, but also provides a ranking between states of multiple state models. Since Russell does not teach or suggest

the use of state models having hierarchial organized states and since each limitation of claim 11 is performed in response to the state models, Russell does not teach or suggest any of the limitations of claim 11. Claim 11 is therefore also novel, nonobvious, and is in a condition for allowance.

Applicants further submit that since claims 2-6, 12-15, and 20 depend from claims 1, 8, and 11, respectively, claims 2-6, 12-15, and 20 are also novel, nonobvious, and are in a condition for allowance for at least the same reasons as put forth above with respect to claims 1, 8, and 11.

Claims 9-10 and 16-19 stand rejected under 35 U.S.C. as being unpatentable over Russell in view of Bernaden III et al. (USPN 6,477,439 B1). Applicants submit that since claims 9-10 and 16-19 depend from claims 8 and 11, respectfully, that claims 9-10 and 16-19 are also novel, nonobvious, and are in a condition for allowance for at least the same reasons as put forth above with respect to claims 8 and 11.

In light of the amendments and remarks, the Applicants submit that all objections and rejections are now overcome. The Applicants have added no new matter to the application by these amendments. The application is now in condition for allowance and expeditious notice thereof is earnestly solicited. Should the Examiner have any questions or comments, he is respectfully requested to call the undersigned attorney.

Respectfully submitted,

ARTZ & ARTZ P.C.

Jeffrey J. Chapp

Jeffrey J. Chapp, Reg. No. 50,579
28333 Telegraph Road, Suite 250
Southfield, MI 48034
(248) 223-9500

Dated: May 5, 2004